# SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

**An Autonomous Institution**

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING-IOT Including CS&BCT

COURSE NAME : 19SB602 FULL STACK DEVELOPMENT FOR NEXT GENERATION IOT

III YEAR / VI SEMESTER

Unit III- **CORE PHP WITH MODEL–VIEW–CONTROLLER**
Topic : Loops, Functions

**PHP Loops**

If you want the same block of code to run over and over again a certain number of times.

So, instead of adding several almost equal code-lines in a script, we can use loops.

Loops are used to execute the same block of code again and again, as long as a certain condition is true.

## Loop Types

while - loops through a block of code as long as the specified condition is true

do...while - loops through a block of code once, and then repeats the loop as long as the specified condition is true

for - loops through a block of code a specified number of times

foreach - loops through a block of code for each element in an array

# PHP while Loop

The while loop - Loops through a block of code as long as the specified condition is true.

The while loop executes a block of code as long as the specified condition is true.

## Example

Print $i as long as $i is less than 6:

```
$i = 1;
while ($i < 6)
{
  echo $i;
  $i++;
}
```

**The break Statement**

With the break statement we can stop the loop even if the condition is still true:

**Example**

Stop the loop when $i is 3:

```
$i = 1;
while ($i < 6) {
  if ($i == 3) break;
  echo $i;
  $i++;
}
```

**PHP do while Loop**

The do...while loop - Loops through a block of code once, and then repeats the loop as long as the specified condition is true.

The do...while loop will always execute the block of code at least once, it will then check the condition, and repeat the loop while the specified condition is true.

**Example**

Print $i as long as $i is less than 6:

```
$i = 1;
do {
  echo $i;
  $i++;
} while ($i < 6);
```

**PHP for Loop**

The for loop - Loops through a block of code a specified number of times.

The for loop is used when you know how many times the script should run.

**Syntax**

for (expression1, expression2, expression3)
{
  // code block
}

**Operations**

expression1 is evaluated once

expression2 is evaluated before each iterarion

expression3 is evaluated after each iterarion

**Example**

Print the numbers from 0 to 10:

```
for ($x = 0; $x <= 10; $x++) {
  echo "The number is: $x <br>";
}
```

# PHP foreach Loop

The foreach loop - Loops through a block of code for each element in an array or each property in an object.

The most common use of the foreach loop, is to loop through the items of an array.

**Example**

Loop through the items of an indexed array:

$colors = array("red", "green", "blue", "yellow");

foreach ($colors as $x) {
  echo "$x <br>";
}

**The foreach Loop on Objects**
The foreach loop can also be used to loop through properties of an object:

**Example**
Print the property names and values of the $myCar object:

```
class Car {
 public $color;
 public $model;
 public function __construct($color, $model) {
   $this->color = $color;
   $this->model = $model;
 }
}
```

```php
$myCar = new Car("red", "Volvo");

foreach ($myCar as $x => $y)

{
  echo "$x: $y <br>";
}
```

## Alternative Syntax

The foreach loop syntax can also be written with the endforeach statement like this

## Example
Loop through the items of an indexed array:

$colors = array("red", "green", "blue", "yellow");

foreach ($colors as $x) :
 echo "$x <br>";
endforeach;

# PHP Functions

The real power of PHP comes from its functions.

PHP has more than 1000 built-in functions, and in addition you can create your own custom functions.

## PHP Built-in Functions

PHP has over 1000 built-in functions that can be called directly, from within a script, to perform a specific task.

Please check out our PHP reference for a complete overview of the PHP built-in functions.

# PHP User Defined Functions

- ✓ Besides the built-in PHP functions, it is possible to create your own functions.

- ✓ A function is a block of statements that can be used repeatedly in a program.
- ✓ A function will not execute automatically when a page loads.
- ✓ A function will be executed by a call to the function.

## Create a Function

A user-defined function declaration starts with the keyword function, followed by the name of the function:

## Example

```
function myMessage() {
 echo "Hello world!";
}
```

**Call a Function**

To call the function, just write its name followed by parentheses ():

**Example**

```
function myMessage() {
  echo "Hello world!";
}


myMessage();
```

## PHP Function Arguments

Information can be passed to functions through arguments. An argument is just like a variable.

Arguments are specified after the function name, inside the parentheses. You can add as many arguments as you want, just separate them with a comma.

**Example**
```
function familyName($fname)
{
  echo "$fname Refsnes.<br>";
}

familyName("Jani");
familyName("Hege");
familyName("Stale");
familyName("Kai Jim");
familyName("Borge");
```

## PHP Default Argument Value

The following example shows how to use a default parameter. If we call the function setHeight() without arguments it takes the default value as argument:

**Example**
```
function setHeight($minheight = 50) {
  echo "The height is : $minheight <br>";
}
setHeight(350);
setHeight(); // will use the default value of 50
setHeight(135);
setHeight(80);
```

Any Query????

Thank you……